CLAIMS

I claim:

1. A method for determining software complexity, comprising the steps of:

determining a plurality of versions of software whose complexity is to be found;

compressing each of the versions, to provide compressed versions;

finding lengths of the compressed versions; and

comparing the lengths of the compressed versions to provide a software complexity metric.

2. The method of claim 1, wherein the plurality of versions includes raw program text.

3. The method of claim 1, wherein the plurality of versions includes normalized program text.

4. The method of claim 1, wherein the plurality of versions includes normalized unique program text.

5. The method of claim 1, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of raw program text and the length of the compressed version of normalized program text.

6. The method of claim 1, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of normalized program text and the length of the compressed version of normalized unique program text.

7. A method for determining software complexity, comprising the steps of:

determining raw program text and normalized program text of software whose complexity is to be found;

compressing the raw program text and the normalized program text to provide compressed raw program text and compressed normalized program text, respectively;

finding the length of the compressed raw program text and the length of the compressed normalized program text; and

finding a ratio using the length of the compressed raw program text and the length of the compressed normalized program text, to provide a complexity metric.

8. A method for determining software complexity, comprising the steps of:

determining normalized program text and normalized unique program text of software whose complexity is to be found;

compressing the normalized program text and the normalized unique program text to provide compressed normalized program text and compressed normalized unique program text, respectively;

finding the length of the compressed normalized program text and the length of the compressed normalized unique program text; and

finding a ratio using the length of the compressed normalized program text and the length of the compressed normalized unique program text, to provide a complexity metric.

9. Apparatus for determining software complexity, comprising:

logic for determining a plurality of versions of software whose complexity is to be determined and for finding lengths of compressed versions of the plurality of versions of the software;

means for compressing each of the versions of the software whose complexity is to be found, to provide the compressed versions; and

means for comparing the lengths of the compressed versions to provide a complexity metric.

10. Apparatus for determining program complexity; comprising:

logic for determining raw program text and normalized program text of software whose complexity is to be determined and for finding the length of compressed raw program text and the length of the compressed normalized program text;

means for compressing the raw program text and the normalized program text to provide the compressed raw program text and the compressed normalized program text, respectively; and means for finding a ratio using the length of the compressed raw program text and the length of the compressed normalized program text, to provide a complexity metric.

11. Apparatus for determining software complexity, comprising:

logic for determining normalized program text and normalized unique program text of software whose complexity is to be determined and for finding the length of compressed normalized program text and the length of the compressed normalized unique program text;

means for compressing the normalized program text and the normalized unique program text to provide the compressed normalized program text and the compressed normalized unique program text, respectively; and

means for finding a ratio using the length of the compressed normalized program text and the length of the compressed normalized unique program text, to provide a complexity metric.

12. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for determining software complexity, said method steps comprising:

determining a plurality of versions of software whose complexity is to be found;

compressing each of the versions, to provide compressed versions;

finding lengths of the compressed versions; and

comparing the lengths of the compressed versions to provide a software complexity metric.

13. The program storage device of claim 12, wherein the plurality of versions includes raw program text.

14. The program storage device of claim 12, wherein the plurality of versions includes normalized program text.

15. The program storage device of claim 12, wherein the plurality of versions includes normalized unique program text.

16. The program storage device of claim 12, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of raw program text and the length of the compressed version of normalized program text.

17. The program storage device of claim 12, wherein the step of comparing includes a step of finding a ratio using the length of the compressed version of normalized program text and the length of the compressed version of normalized unique program text.

18. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for determining software complexity, said method steps comprising:

determining raw program text and normalized program text of software whose complexity is to be found;

compressing the raw program text and the normalized program text to provide compressed raw program text and compressed normalized program text, respectively;

finding the length of the compressed raw program text and the length of the compressed normalized program text; and

finding a ratio using the length of the compressed raw program text and the length of the compressed normalized program text, to provide a complexity metric.

19. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for determining software complexity, said method steps comprising:

determining normalized program text and normalized unique program text of software whose complexity is to be found;

compressing the normalized program text and the normalized unique program text to provide compressed normalized program text and compressed normalized unique program text, respectively;

finding the length of the compressed normalized program text and the length of the compressed normalized unique program text; and

finding a ratio using the length of the compressed normalized program text and the length of the compressed normalized unique program text, to provide a complexity metric.